System for simulating high bitcoin transaction volumes

BITS ZG628T: Dissertation

By

M A Javed Khan

(2013HT13062)

Dissertation work carried out at

Agiliq

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI (RAJASTHAN)

October 2015

System for simulating high bitcoin transaction volumes

BITS ZG628T: Dissertation

By

M A Javed Khan

(2013HT13062)

Dissertation work carried out at

Agiliq, Hyderabad

Submitted in partial fulfillment of M.Tech. Software Systems degree programme Under the Supervision of Akshar Raaj, Team Lead, Agiliq Info Solutions, Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)

October 2015

CERTIFICATE

This is to certify that the Dissertation entitled ***System for simulating high bitcoin transaction volumes*** *and submitted by* ***M A Javed Khan*** *having ID No.* ***2013HT13062*** *for the partial fulfilment of the requirements of M.Tech. Software Systems degree of BITS Pilani, embodies the bonafide work done by him under my supervision.*

Signature of the Supervisor

Place**: **Hyderabad

Date**: **30.10.2015

Akshar Raaj, Team Lead, Agiliq

(Name, Designation & Organization & Location)

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

WORK-INTEGRATED LEARNING PROGRAMMES DIVISION

First Semester 2015-2016

BITS ZG628T: Dissertation

Abstract

ID No.	**:**	2013HT13062
NAME OF THE STUDENT	**:**	M A Javed Khan
EMAIL ADDRESS	**:**	javed.ma@gmail.com
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	**:**	Agiliq Info Solutions, Hyderabad
SUPERVISOR'S NAME	**:**	Akshar Raaj
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	**:**	Agiliq Info Solutions, Hyderabad
DISSERTATION TITLE	**:**	System for simulating high bitcoin transaction volumes

Abstract

Bitcoin is a decentralized digital currency that enables instant payments to anyone, anywhere in the world. It uses peer-to-peer technology to operate with no central authority: transaction management and money issuance are carried out collectively by the network.

Today the Bitcoin network is restricted to a sustained rate of 7 transactions per second due to the bitcoin protocol restricting block sizes to 1MB. The issue of block size and scalability of bitcoin as the blockchain size increases in the future is a widely debated topic.

In this work, we develop a method for stress testing core parts of the bitcoin architecture by simulating high volumes of transactions. By running such simulations we can answer several questions that can help in scaling bitcoin, such as the practical limit of how many transactions can be included per block, how many transactions can be processed per second, what are the benchmarks for CPU and memory performance, issues and failure conditions during stress testing etc.

Broad Academic Area of work: Software Testing and Quality Assurance

Keywords: Bitcoin, Cryptocurrency, Simulation Testing

Signature of the student	**Signature of the supervisor**
Name: M A Javed Khan	**Name: Akshar Raaj**
Date: 30/Oct/2015	**Date: 30/Oct/2015**
Place: Hyderabad	**Place: Hyderabad**

Acknowledgements

I take this opportunity to express my sincere thanks and regards to my supervisor, Akshar Raaj, Team Lead, Agiliq for his encouragement, and support throughout the course of this Dissertation.

I am grateful to Shabda Raaj, Director, Agiliq for giving me the opportunity to work on this topic.

This work would not have been possible without the help and guidance of Jacob-Yocom Piatt, Dave Collins, Josh Rickmar, and the btcsuite team, so I appreciate their mentorship and kindness.

M A Javed Khan

Table Of Contents

Table Of figures

Glossary

- TPS/tps:...................................Transactions per second
- TPB/tpb:..................................Transactions per block
- RPC:..Remote Procedure Call
- ECDSA:....................................Elliptic Curve Digital Signature Algorithm

1. Introduction:

Bitcoin is a decentralized digital currency that enables instant payments to anyone, anywhere in the world. Bitcoin uses peer-to-peer technology to operate with no central authority: transaction management and money issuance are carried out collectively by the network. The original Bitcoin software by Satoshi Nakamoto was released under the MIT license. Most client software, derived or "from scratch", also use open source licensing [1]. Bitcoin is the first successful implementation of a distributed crypto-currency, described in part in 1998 by Wei Dai on the cypherpunks mailing list. Building upon the notion that money is any object, or any sort of record, accepted as payment for goods and services and repayment of debts in a given country or socio-economic context, Bitcoin is designed around the idea of using cryptography to control the creation and transfer of money, rather than relying on central authorities [2].

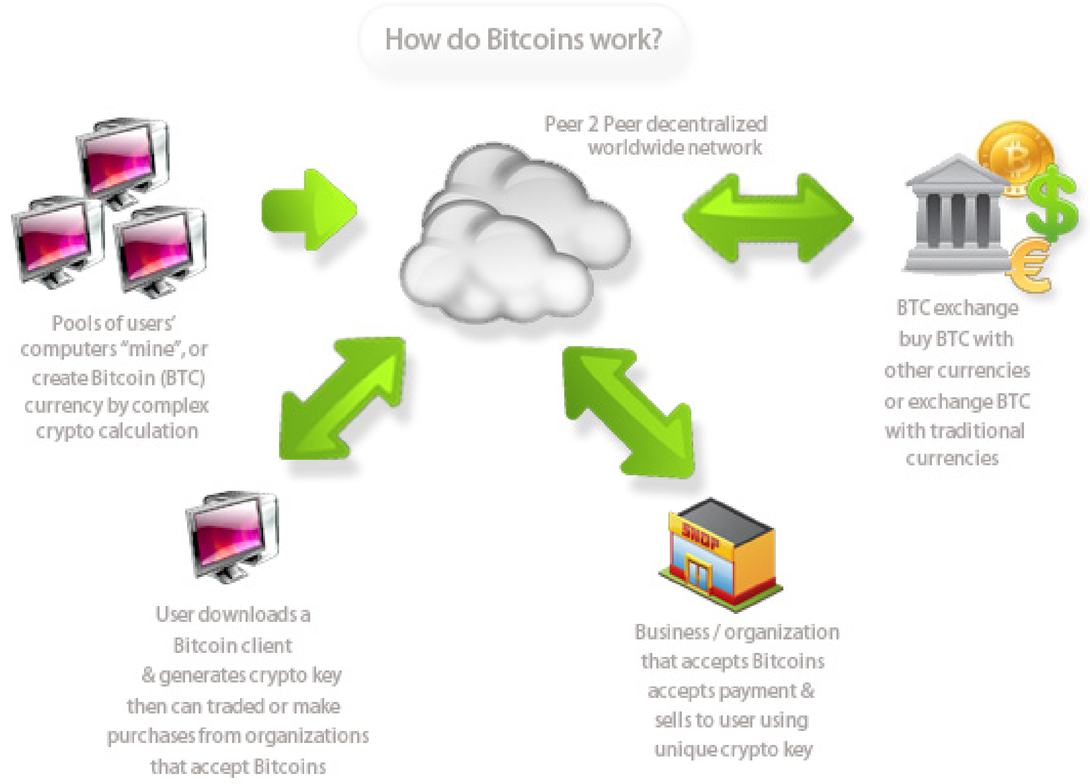
How do Bitcoins work?
Peer 2 Peer decentralized worldwide network
Pools of users' computers "mine", or create Bitcoin (BTC) currency by complex crypto calculation
BTC exchange buy BTC with other currencies or exchange BTC with traditional currencies
User downloads a Bitcoin client & generates crypto key then can traded or make purchases from organizations that accept Bitcoins
SHOP
Business / organization that accepts Bitcoins accepts payment & sells to user using unique crypto key

Figure 1: How do Bitcoins work

2. Background:

Previous work in this field was conducted by the btcsuite team using `btcsim`[3], which was comprehensive, however it was limited to `btcd`, the alternate implementation in Go. In this Dissertation we conducted similar simulations on `bitcoind`[4], the official C++ implementation.

3. Objectives:

The idea is to simulate load on the Bitcoin network and see how the existing full nodes and wallets behave under that load. This will allow us to answer many questions whose answers are currently the subject of speculation:

- What is the maximum practical limit for full nodes in terms of average transactions per second ("tps"), transactions per block ("tpb")?
- Which parts of the existing network infrastructure fail or have problems under what conditions?
- How many tps can be modeled using a single machine with one or more "agents" creating simulated transactions on it?

4. Scope of work:

As a part of this project, a test driver will be implemented which simulates transaction volumes and measures the throughput of the network. Behavior of nodes with respect to various transaction rates per block will be explored and discussed. The implementation will be assessed based on the following milestones:

- A 1-agent simulation where the wallet contains 1000 addresses and the agent only sends payments to addresses it already controls.
- A N-agent simulation where each wallet contains 1000 addresses and the agents randomly send funds to each others addresses.
- A N-agent simulation with 1000 addresses each, which takes a curve for average transaction rate as an input.

5. Architecture:

For this project, we use the binary application `bitcoind` which acts as a bitcoin node for the simulation. The simulation launches the node, initializes the agents, manages their addresses and sends funds as payments between them. After the simulation is done, a report of the statistics will be displayed.

To be able to create transactions without using real bitcoins, we use the regtest mode [5] provided by `bitcoind`. It allows us to generate mock bitcoins instantly without being connected as a peer to the bitcoin network.

We break down the application into the following components:

a. Test Driver:

The test driver launches the simulation, initializes agents, manages them using a RPC client, and generates transactions. Once done, it cleans up by gracefully shutting down processes and reports the statistics.

b. Actor:

An Actor simulates the agent by launching a node which acts as wallet agent and interacts with the test driver to generate transactions. The test driver may initialize one or more actors to run an N-agent simulation.

c. RPC Client:

For the Test Driver to communicate with the Actor, we need a RPC client. The client manages the connection to the Actor and sends a request to create a transaction.

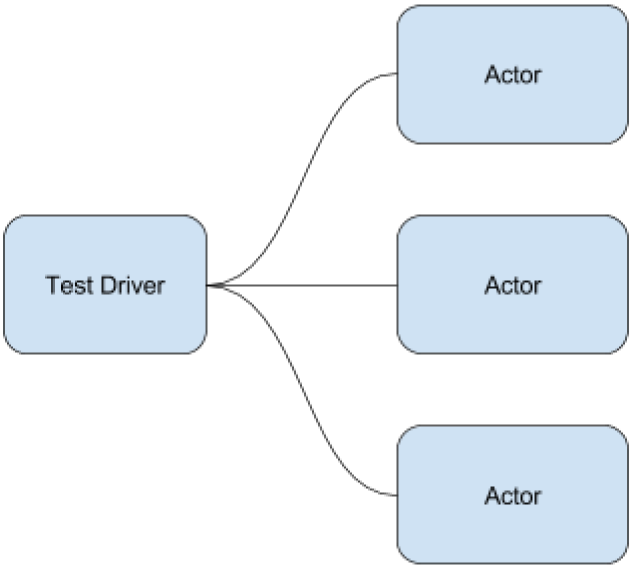

Figure 2: Architecture

This architecture (as shown in Figure 2) enables the simulation of a blockchain with very large block sizes (up to 32MB), starting at a small block height with a minimal set of components involved.

6. Simulation:

The default block size in `bitcoind` is 75000 bytes, therefore we need to modify and set `blockmaxsize` flag on the command line to 32 MB (`--blockmaxsize=33554432`). With this simple configuration change, we can simulate blocks larger than the default protocol limit of 1 MB. We chose 32 MB as the block size for this simulation since the protocol limits all P2P messages to 32 MB.

Note that the following simulations were run on a average hardware CPU miner. The results for proper mining hardware could be different.

We simulated a blockchain with 180,000 transactions per block. The process of setting up the transactions takes nearly 12-15 hours depending on the hardware. For mining the blocks, we found that an average CPU takes 50-80 minutes per block. A 32 MB block can hold a maximum of approximately 167,000 simple P2PKH transactions, so the remaining transactions will remain in the memory (mempool). Assuming, the block generation time could be optimized to 10 minutes, we can scale the number of transactions per second to an average of 267 tps, which is far greater than the current average of 0.9 tps on the main network (mainnet).

7. Summary:

We found that the `bitcoind` implementation was also capable of handling far larger block sizes than the current average.

We did not experience any failure modes or issues during the stress testing.

The time taken for generating large blocks was found to be significantly greater in `bitcoind`, when compared to `btcd`.

8. Conclusions and Recommendations:

The simulation results show that larger block sizes can be handled, although optimizations might be required to ensure timely generation of blocks. In particular, this might involve a architecture with load-balancing approach to costly operations like ECDSA signature verification.

9. Directions for future work:

We hope that the simulation results will pave way for more rigorous and standardized benchmarks, with analysis of performance providing insights into optimizations which can help the scalability of bitcoin.

10. References:

1) Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." *Consulted* 1.2012 (2008): 28.
2) Grinberg, Reuben. "Bitcoin: an innovative alternative digital currency." *Hastings Sci. & Tech. LJ* 4 (2012): 159.
3) https://blog.conformal.com/btcsim-simulating-the-rise-of-bitcoin/
4) https://github.com/bitcoin/bitcoin
5) https://bitcoin.org/en/developer-examples#regtest-mode

Checklist of items in the report

This checklist is to be duly completed, verified and signed by the student.

1.	**Is the final report neatly formatted with all the elements required for a technical Report?**	Yes
2.	Is the Cover page in proper format as given in Annexure A?	Yes
3.	Is the Title page (Inner cover page) in proper format?	Yes
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes Yes
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes Yes
6.	Is the title of your report appropriate? **The title should be adequately descriptive, precise and must reflect scope of the actual work done.** Uncommon abbreviations / Acronyms should not be used in the title	Yes
7.	Have you included the List of abbreviations / Acronyms?	Yes
8.	Does the Report contain a summary of the literature survey?	Yes
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate	Yes Yes Yes Yes Yes Yes
10.	Is the conclusion of the Report based on discussion of the work?	Yes
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes Yes Yes
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	Yes

Declaration by the student

I certify that I have properly verified all the items in the checklist and ensure that the report is in proper format as specified in the course handout.

Place: **Hyderabad**	**Signature of the student**
Date: **30-10-2015**	**Name: M A Javed Khan**
	ID No: 2013HT13062